

**REMARKS**

In the non-final Office Action, the Examiner rejects claims 39-41, 45-49, 53, 54, 57-60, 63-68 and 71 under 35 U.S.C. § 103(a) as being unpatentable over CHIAPPA (U.S. Patent No. 5,249,292) in view of MORRISSEY et al. (U.S. Patent No. 5,463,762); and rejects claims 42-44, 51, 52, 55, 56, 61, 62, 69, 70 and 72 under 35 U.S.C. § 103(a) as being unpatentable over CHIAPPA in view of MORRISSEY et al., and further in view of SUZUKI (U.S. Patent No. 4,799,215). Applicants respectfully traverse these rejections.

Claims 39-49 and 51-72 remain pending in the present application. Timely reconsideration and allowance of all claims in view of the following remarks are respectfully requested.

**Rejections under 35 U.S.C. § 103(a)**

Claims 39-41, 45-49, 53, 54, 57-60, 63-68 and 71 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over CHIAPPA in view of MORRISSEY et al. This rejection is respectfully traversed.

A proper rejection under 35 U.S.C. § 103 requires that three basic criteria be met. First, there must be some suggestion or motivation, either in the references themselves, or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest each and every claim limitation. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not the applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). The cited combination of CHIAPPA

and MORRISSEY et al. fail to disclose or reasonably suggest the combination of features recited in claims 39-41, 45-50, 53, 54, 57-60, 63-68 and 71.

Claim 39, for example, is directed to an apparatus for processing packets. The apparatus comprises a first input queue configured to receive a stream of incoming packets and to output beginning portions of packets as the beginning portions are received without waiting for the respective packets to be received in their entirety; a first in-line packet processor for receiving the beginning portions from the first input queue, each beginning portion including first header information, and for detecting the existence of an error in the first header information of each beginning portion; and a first memory for storing packets received at the first input queue and for which the first in-line packet processor did not detect an error in the corresponding first header information.

The cited combination of CHIAPPA and MORRISSEY et al. do not disclose or suggest the combination of features recited in claim 39. For example, neither CHIAPPA nor MORRISSEY et al. disclose or suggest a first in-line packet processor for receiving the beginning portions from the first input queue, and for detecting the existence of an error in the first header information of each beginning portion. In making the rejection, the Examiner alleged that flow block 14a and pattern matcher 16a in CHIAPPA correspond to the above-recited in-line packet processor of claim 39 and that input buffer 26 corresponds to the above-recited first input queue (Office Action, pp. 2-3). The Examiner further alleged that col. 3, line 62 to col. 4, line 12, col. 7, line 46 to col. 8, line 24, and col. 13, line 55 to col. 14, line 44 of CHIAPPA support these allegations (Id.). Applicants respectfully disagree with the Examiner's interpretation of these portions of CHIAPPA.

At col. 3 line 62 to col. 4, line 12, CHIAPPA discloses:

In particular aspects of the invention, the data stream control circuit features a pattern matching circuit, responsive to pattern setting signals from the primary processing unit and to the incoming data packets from the network interface units, for identifying those packets of a packet stream which will be processed by the control circuit. The data stream control circuit further features a processing unit responsive control circuit for controlling, in response to control signals sent by the primary processing unit, the congestion control and header modification, stripping and prepending functions of the data stream control circuit. The data stream control circuit further features a data buffer responsive to the pattern matching circuitry and the processing unit responsive control circuit for storing data and protocol elements of an incoming data packet stream and for outputting a data packet stream to be forwarded along a communications path.

This section of CHIAPPA discloses a data stream control circuit including a pattern matcher and a buffer for identifying and storing packets in a received data stream. This section of CHIAPPA does not disclose an in-line packet processor for receiving the beginning portions from the first input queue, each beginning portion including first header information, and for detecting the existence of an error in the first header information of each beginning portion, as recited in claim 39. In fact, this section of CHIAPPA does not relate to header reception or error detection whatsoever.

At col. 7, line 46 to col. 8, line 24, CHIAPPA discloses:

In operation, a traffic stream is received and first identified by the CPU 12, as it receives the first packet of a new traffic stream from a CPU input buffer 26 connected to the input interconnect path 31. A free flow block 14 is selected to handle future packets of that traffic stream and all of the necessary information to handle the traffic stream, including the identification of the stream, is loaded into the pattern matching circuitry 16 and the control circuitry 18 of the selected flow block over the CPU bus 41.

As each subsequent packet of the stream arrives at the packet switch interface circuit, it is handled by the network interface 30 (for ease of explanation it is generally assumed that the receiving network device will be an interface 30) and flow block 14 without intervention by the CPU 12. In particular, as it is received at interface circuit 30, it passes through the network interface circuitry 30 and is placed on the input interconnect path 31 so that each flow block 14, assigned to that interface, can check the packet, in parallel, to determine if any one of those flow blocks recognizes the packet as being assigned to it. If

a match is found, the packet is accepted by that flow block and the data, usually modified by the control circuitry 18 of the flow block, is read and stored by the flow block. Further circuitry of control circuitry 18 will remove the packet from the data buffer 20 of the flow block 14, with a new header prepended thereto, when the system is ready to send the packet over the next link of the data communications path.

Any packet which is not recognized by any of the flow blocks is available to the CPU from the one of the CPU input buffers 26 assigned for receiving data from that network interface. The CPU input buffer for each network automatically starts to copy each packet from the input interconnect path 31 each time a packet arrives, and continues to do so until one of the flow blocks 14 for that network interface accepts, or all flow blocks assigned to that network interface reject, the packet. If the packet was accepted by one of the assigned flow block circuitries, the portion of the data stored in the associated CPU input buffer 26 is discarded, and the CPU input buffer resets to await the next packet from that network interface. If the packet is rejected by those flow blocks assigned to that network interface, the associated buffer 26 passes the packet to the processor 12 which will analyze the packet and process it accordingly.

This section of CHIAPPA discloses that a received packet is checked in parallel by all available flow blocks 14 to determine to which flow block the packet is assigned. Simultaneously, the packet is copied to the input buffer 26 to cover the possibility that no flow block is assigned to the received stream. In this situation, an available flow block is assigned to the stream and all subsequently received packets belonging to that stream will be handled by the assigned flow block. This section of CHIAPPA does not disclose an in-line packet processor for receiving the beginning portions from the first input queue, each beginning portion including first header information, and for detecting the existence of an error in the first header information of each beginning portion, as recited in claim 39. On the contrary, this section of CHIAPPA discloses that once assigned to a flow block, all received packets are processed and stored directly by the flow block, without storage in buffer 26. Accordingly, even assuming, *arguendo*, that flow block 14 and pattern matcher 16 correspond to an in-line packet processor (a point that Applicants do not concede), CHIAPPA clearly fails to disclose forwarding beginning portions of

received packets from the first input queue, as required by claim 39.

Moreover, this section of CHIAPPA likewise fails to disclose or even remotely suggest an in-line packet process for detecting the existence of an error in the first header information of each beginning portion, as recited in claim 39. In fact, this section of CHIAPPA makes no reference to error detection whatsoever.

At col. 13, line 55 to col. 14, line 44, CHIAPPA discloses:

Referring to FIG. 4, the circuit structure of flow block 14, considered in more detail, has an input multiplexor 250 which selects the current input bus and passes the data to both the pattern matcher 16 and the rest of the flow block. The pattern matcher, as noted above, examines the header of the incoming packet. If it matches the pattern to be handled by this flow block, the match is indicated by a signal over a line 252 to the control device logic 18.

Simultaneously, data from the input bus flows through a stripping circuit 254 which includes a counter and which discards the first "n" bytes of data (the header) allowing the remainder of the packet to pass through unmodified. The packet then passes to the control logic 18 where the higher level protocol functions such as check sum computation and hop count modification occur. The control logic 18, pattern matcher 16, and stripping circuit 254 have all been previously loaded with other necessary data from CPU 12 over bus 41. The input to the control device has a small amount of buffering to allow the control device to take more than one cycle when processing certain bytes in the data stream. The packet passing through this stage of processing may be modified; for example, this stage may abort further processing of the packet if an error is found, as described in more detail below. The packet then passes to a counter/truncate circuitry 260 which contains a counter loaded by the control logic over circuitry 262. The counter serves two functions: any unused trailer in the packet is discarded, and, if the packet is truncated, an error flag is raised over a line 264. The next stage of processing, a circuitry 266, prepends "n" bytes of data, the new output header, loaded from the CPU 12 in a similar manner to stripping circuit 254, to the packet as it passes therethrough. It also contains some buffering on the input to allow the new packet header to be inserted. In those instances where the new packet is substantially larger than the old one, the buffering is a necessity. The packet next passes to the output data buffer 20 which consists of a dual port (one read-only and one write-only) memory, along with a control logic 268 to keep track of the packets in the buffer. The buffer 20 is organized in a ring structure and a hardware queue of "t" buffer pointer/size pairs keeps track of the utilization of the buffer. Additional control circuitry within the buffer keeps track of the current start and end of the "free space". The packet then passes to an output multiplexor 274 which has output bus control logic and a set of drivers, one for each output bus in the

output interconnect 52. When the flow block receives the "grant," for the appropriate output network interface 30, as described above, packets which are in the output buffer are read out and passed along the bus. Throughout the flow block, there are, in addition, data paths 276 which allow the CPU 12, over bus 41, to load memories, etc. in order to maintain proper operation of the flow block.

This section of CHIAPPA discloses the electronic circuit configuration of each flow block 14. In particular, it is disclosed that packet headers are checked to determine whether the flow block is the correct flow block to handle the packet. Once matched to a flow block, the received packets are stripped of their headers and new headers are generated and inserted prior to storing the received packet in an output buffer 20. This section of CHIAPPA does not disclose or suggest an in-line packet processor for receiving the beginning portions from the first input queue, each beginning portion including first header information, and for detecting the existence of an error in the first header information of each beginning portion, as recited in claim 39.

The disclosure of MORRISSEY et al. does not remedy the above-noted deficiencies in the disclosure of CHIAPPA. For at least this reason claim 39 is patentable over the cited combination of CHIAPPA and MORRISSEY et al.

Moreover, the combination of CHIAPPA and MORRISSEY et al. also does not disclose or suggest a first input queue configured to receive a stream of incoming packets and to output beginning portions of packets as the beginning portions are received without waiting for the respective packets to be received in their entirety, as recited in claim 39. The Examiner acknowledged that CHIAPPA does not disclose this feature (Office Action, pg. 3). To remedy this deficiency, the Examiner relied on col. 5, line 44 to col. 6, line 18, col. 20, lines 41-57, and col. 21, line 44 to col. 22, line 34 of MORRISSEY et al. for allegedly disclosing instantaneous packet processing by forwarding header portions of packets to an in-line packet processor

without waiting for the entire packet to be received (Id.). The Examiner then alleged that such a disclosure may be easily adopted by one of ordinary skill in the art to implement into the system of CHIAPPA to further improve the system reliability and efficiency. Applicants respectfully traverse.

At col. 5, line 44 to col. 6, line 18, MORRISSEY et al. discloses:

FIG. 3 is a block diagram of an exemplary receive function according to the invention. The receive function is architecturally similar to the transmit function, in that respectively different elements 82-86 are provided for processing incoming control data. Items 82-86 are described in greater detail below with reference to FIGS. 10-13. Sequence recognition logic 82 processes link level ordered sequences (e.g., idles) to identify a change of the state of channel 20. An SOF recognition function 83 detects the receipt of an SOF delimiter. A receive frame buffer 84 stores an incoming frame header, so that the header may be verified. A CRC check function 85 verifies the user data within the frame. The EOF recognition function 86 identifies an EOF delimiter which indicates the end of the frame.

Receive switching facility 81 is analogous to the transmit switching facility 80. Switching facility 81 includes logic to determine when to switch the incoming data stream from one of the processing functions 82-86 to another, so that each portion of the data stream is processed correctly.

According to the invention, an expected type of frame is determined, and the data in the frame buffer 84 are compared to the expected type information, to determine whether the type of frame received is an "expected type" of frame, e.g., complying with the link and device level I/O protocols.

By providing hardware elements to generate control data and perform the switching between the user defined data and the control data, microprocessor interrupts are reduced and the microprocessor resource is freed up to direct the high level functions and respond to error conditions. Similarly, by using separate hardware elements to process received control data and to direct the switching of the control data to the hardware, efficient use is made of the microprocessor 52.

According to another aspect of the invention, shown in FIG. 9, distinct elements 354 and 356 are provided for asynchronously forming device block header data and error detection code (EDC) data, respectively. Control elements 352, 362 and 366 inject these data into the received data stream en route to the data buffer 38 (shown in FIG. 3). In the transmit path 31, the control elements 352, 362 and 366 strip the header and EDC out of the data stream before transmission over the serial medium 36. Element 354 checks the

block header and element 356 checks EDC data.

This section of MORRISSEY et al. discloses receiving and examining control data. More specifically, this section of MORRISSEY et al. discloses recognizing a start of frame (SOF) delimiter, storing an incoming frame header in a receive frame buffer 84, so that the header may be verified, performing a CRC check function to verify the data, and recognizing an end of frame (EOF) delimiter. This section of MORRISSEY et al. does not disclose or suggest a first input queue configured to receive a stream of incoming packets and to output beginning portions of packets as the beginning portions are received without waiting for the respective packets to be received in their entirety, as recited in claim 39. In fact, no disclosure is made whatsoever regarding the amount of data received, but merely to the processing performed on received data.

At col. 20, lines 41-57, MORRISSEY et al. discloses:

FIGS. 10-12 are detailed block diagrams of the receive logic hardware 29 of the fiber interface 26. Referring first to FIG. 10, the fiber interface receive logic 29 includes the sequence receive logic 82, frame recognition logic 404, CRC checker 85, frame counter 408 and violation/realign logic 410. The receive logic 29 receives, strips out and processes the control data in each frame.

The frame receiving logic 29 receives and identifies a frame header from the data transfer medium 36. According to the invention, frame receiving logic 29 and microprocessor 52 verify link and device level I/O protocol compliance asynchronously while any user defined data that are included in the frame are being received, and store the user defined data in the parallel storage medium 38, if any user defined data are included. Frame receiving logic 29 performs a CRC check and captures the type of SOF and type of EOF and stores them in the frame status bytes 452 of a frame buffer 84.

This section of MORRISSEY et al. discloses an optical fiber interface for receiving data frames and performing processing on the received frames. Further, this section of MORRISSEY et al. discloses detecting verify link and protocol compliance while user data within the frame is being received. This section of MORRISSEY et al. does not disclose or suggest a first input queue



configured to receive a stream of incoming packets and to output beginning portions of packets as the beginning portions are received without waiting for the respective packets to be received in their entirety, as recited in claim 39. In fact MORRISSEY et al. does not disclose packet reception or processing at all, but instead relates to hardware-level frame reception.

At col. 21, line 44 to col. 22, line 34, MORRISSEY et al. discloses:

In the exemplary embodiment, the SOF recognition and EOF recognition functions 83 and 86 (shown in FIG. 3) are both included within the frame recognition logic 404, as shown in FIG. 10. The frame recognition logic 404 strips the SOF and EOF characters from the frame of data. In a normal data stream, the frame recognition logic 404 detects SOF-EOF pairs. When the SOF recognition function 83 of frame recognition logic 404 detects an SOF, it transmits a frame-in-progress signal to frame counter 408, CRC checker 85 and the receive frame buffer switch control 450 (shown in FIG. 11). The frame recognition logic 404 then waits until the EOF recognition function 86 detects an EOF character. An identification of the type of SOF and type of EOF in the pair is transmitted to the frame status flag bytes 452 of the receive frame buffer 84.

If frame recognition logic 404 detects two SOF characters without an intervening EOF character, or an SOF character followed by any special ("K") character without an intervening EOF character, then frame recognition discontinues the frame in progress signal 422 and transmits an error signal 424 to microprocessor 52.

In the exemplary embodiment, frame recognition logic 404 implements the rules for defining a frame set forth in chapter 2 of the ESCON I/O Interface specification, referenced above. One skilled in the art can readily construct the frame recognition logic from a plurality of registers for storing the SOF and EOF characters, and comparators for checking the incoming characters in the data stream to detect the presence of SOF or EOF characters.

While the frame in progress signal 422 is transmitted by the frame recognition logic 404, the frame counter 408 begins to count the number of valid bytes in the frame. An AND gate 409 transmits a signal that increments the frame counter 408 when the frame in progress signal 422 is in its active state and the valid bit of the received data is set. The count of bytes in the frame is transmitted by frame counter 408 to the frame byte count bytes 454 of the receive frame buffer 84 (shown in FIG. 12). The frame counter 408 is reset when the frame in progress signal 422 changes to its inactive state.

When the frame in progress signal 422 is first sent, the CRC checker 85 begins to compute a CRC based on the data in the frame and generates a parity bit for every byte, which is merged with the byte stream in block 428. When the EOF is detected, by

definition, the last two bytes before the EOF are the CRC of the incoming data stream. Delay elements 426a and 426b delay the stream of data to the CRC checker 85 by the amount of time it takes to receive two bytes, so that when the frame in progress signal 422 stops, CRC checker 85 recognizes the next two bytes it receives as CRC. The data stream is delayed in block 430, while the result of the CRC check is provided to the frame status bytes 452 of the receive frame buffer (shown in FIG. 12). If the CRC checker 85 detects an error, an error signal is provided to the frame status bytes 452 of the receive frame buffer 84 (shown in FIG. 12). The CRC checker 85 also strips the CRC bytes from the frame.

This section of MORRISSEY et al. discloses receiving frames and examining the received frame for its SOF and EOF values. Once received, the frame may be verified by CRC processing. This section of MORRISSEY et al. does not disclose or suggest a first input queue configured to receive a stream of incoming packets and to output beginning portions of packets as the beginning portions are received without waiting for the respective packets to be received in their entirety, as recited in claim 39. In fact, this section of MORRISSEY et al. does not disclose packet reception or processing at all. Furthermore, in relation to frame reception, this section of MORRISSEY et al. discloses receiving an entire frame (as delimited by its SOF and EOF delimiters) prior to performing CRC verification. Clearly, the reception and processing of entire frames does not fairly suggest the outputting of beginning portions of received packets from a first input queue as they are received without waiting for the respective packets to be received, as required by claim 39.

For at least these additional reasons, claim 39 is patentable over the combination of CHIAPPA and MORRISSEY et al. Reconsideration and withdrawal of the pending rejection are respectfully requested.

Even assuming, for the sake of argument, that the above section of MORRISSEY et al. can reasonably be construed to disclose outputting of beginning portions of received packets

from a first input queue as they are received without waiting for the respective packets to be received, as required by claim 39, Applicants submit that one skilled in the art at the time of Applicants' invention would not have been motivated to combine this alleged teaching of MORRISSEY et al. with the CHIAPPA system, absent impermissible hindsight.

With respect to motivation, the Examiner alleges that the disclosure of MORRISSEY et al. may be easily adopted by one of ordinary skill in the art to implement into the system of CHIAPPA to further improve the system reliability and efficiency (Office Action, pg. 3). Applicants respectfully disagree.

The Examiner does not logically explain how incorporating the frame reception system of MORRISSEY et al. into the packet processing system of CHIAPPA would improve the efficiency and reliability of the CHIAPPA system, particularly in light of the fact that the advantages of MORRISSEY are specifically reflected in terms of frame processing. Applicants submit that the Examiner's motivation is merely a conclusory statement regarding an alleged benefit of the combination. Such motivation is insufficient for establishing a *prima facie* case of obviousness.

For at least these additional reasons, Applicants submit that claim 39 is patentable over CHIAPPA and MORRISSEY et al., whether taken alone or in any reasonable combination.

Claims 40, 41, and 45-48 depend from claim 39 and are therefore patentable over the combination of CHIAPPA and MORRISSEY et al. for at least the reasons given above, with respect to claim 39.

Independent claims 49, 53, 59, and 65 recite features similar to (but possibly different in scope than) those described above, with respect to claim 39. Accordingly, claims 49, 53, 59, and

65 are patentable over the combination of CHIAPPA and MORRISSEY et al. for at least reasons similar to those set forth above, with respect to claim 39.

Claims 54, 57, and 58 depend from claim 53 and are therefore patentable over CHIAPPA and MORRISSEY et al. for at least the reasons set forth above, with respect to claim 53.

Claims 60, 63, and 64 depend from claim 59 and are therefore patentable over CHIAPPA and MORRISSEY et al. for at least the reasons set forth above, with respect to claim 59.

Claims 66-68 and 71 depend from claim 65 and are therefore patentable over CHIAPPA and MORRISSEY et al. for at least the reasons set forth above, with respect to claim 65.

Accordingly, Applicants respectfully request the withdrawal of the pending rejection and timely allowance of claims 39-41, 45-49, 53, 54, 57-60, 63-68 and 71 in view of CHIAPPA and MORRISSEY et al.

Claims 42-44, 51, 52, 55, 56, 61, 62, 69, 70, and 72 stand rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over CHIAPPA in view of MORRISSEY et al. and further in view of SUZUKI. Applicants respectfully traverse the rejection.

Claims 42-44 and 72 depend from claim 39, claims 51 and 52 depend from claim 49, claims 55 and 56 depend from claim 53, claims 61 and 62 depend from claim 59, and claims 69 and 70 depend from claim 65. Without acquiescing in the Examiner's rejection, Applicants respectfully submit that the disclosure of SUZUKI does not cure the deficiencies in the disclosures of CHIAPPA and MORRISSEY et al. identified above with regard to claims 39, 49, 53, 59, and 65. Therefore, claims 42-44, 51, 52, 55, 56, 61, 62, 69, 70, and 72 are patentable over CHIAPPA, MORRISSEY et al., and SUZUKI, whether taken alone or in any reasonable combination, for at least the reasons given with regard to claims 39, 49, 53, 59, and 65.

For at least the foregoing reasons, Applicants respectfully request the reconsideration and withdrawal of the rejection of claims 42-44, 51, 52, 55, 56, 61, 62, 69, 70, and 72 under 35 U.S.C. § 103 based on CHIAPPA, MORRISSEY et al., and SUZUKI.

**CONCLUSION**

In view of the foregoing remarks, Applicants respectfully request the Examiner's reconsideration of this application, and the timely allowance of the pending claims.

While the present application is now believed to be in condition for allowance, should the Examiner find some issue to remain unresolved, or should any new issues arise which could be eliminated through discussions with Applicants' representative, then the Examiner is invited to contact the undersigned by telephone in order that the further prosecution of this application can thereby be expedited.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 05-1070 and please credit any excess fees to such deposit account.

Respectfully submitted,

HARRITY SNYDER, L.L.P.

By: /Robin C. Clark/

Robin C. Clark

Reg. No.40,956

Date: August 1, 2006  
11350 Random Hills Road  
Suite 600  
Fairfax, Virginia 22030  
(571) 432-0800